基于改进人工蜂群算法与 MapReduce 的大数据聚类算法 *

孙 倩[a]，陈 昊[b]，李 超[a]

(湖北大学 a.信息化建设与管理处; b.计算机与信息工程学院，武汉 430062)

摘 要：针对大数据聚类算法计算效率与聚类性能较低的问题，提出了一种基于改进人工蜂群算法与 MapReduce 的大数据聚类算法。将灰狼优化算法与人工蜂群算法结合，同时提高人工蜂群算法的搜索能力与开发能力。该策略能够有效地提高聚类处理的性能。采用混沌映射与反向学习作为 ABC 种群的初始化策略，提高搜索的解质量。将聚类算法基于 Hadoop 的 MapReduce 编程模型实现，通过最小化类内距离的平方之和，实现对大数据的聚类处理。实验结果表明，该算法有效地提高了大数据集的聚类质量，同时加快了聚类速度。

关键词：数据分析；聚类算法；人工蜂群算法；灰狼优化算法；云计算；分布式计算

中图分类号：TP391 doi: 10.19734/j.issn.1001-3695.2018.11.0865

Clustering algorithm of big data based on improved artificial bee colony algorithm and MapReduce

Sun Qian[a], Chen Hao[b], Li Chao[a]

(a. *Informationization Management Department*, b. *School of Computer Science & Information Engineering*, *Hubei University*, *Wuhan* 430062, *China*)

Abstract: Aiming at the problems of low computational efficiency and low clustering performance of clustering algorithms for big data, a clustering algorithm of big data based on the improved artificial bee colony algorithm and MapReduce is presented. The grey wolf optimizer algorithm and artificial bee colony algorithm are combined, in order to improve the exploration and exploitation of the artificial bee colony algorithm simultaneously, this strategy helps to improve the clustering performance effectively. The chaotic map and backward learning are utilized as the initial strategy of ABC colony to improve the solution quality of search procedure. The clustering algorithm is realized based on MapReduce programming model, and the clustering process for big data is realized by minimizing the quadratic sum of inner class distances. Experimental results demonstrated that the proposed algorithm improves the clustering quality of big data, and it speedups the clustering procedure.

Key words: data analysis; clustering algorithm; artificial bee colony algorithm; grey wolf algorithm; cloud computing; distributed computing

0 引言

聚类处理是数据分析领域中的一个重要步骤，优质的数据分析是决定决策系统性能的关键因素[1]。许多研究人员提出了有效的聚类算法，但这些聚类算法的计算时间大多随着数据集规模的增大而增加，所以大数据集聚类处理的时间效率是一个关键的性能指标[2]。MapReduce 是一个强大的编程模型，支持并行、分布式地处理大数据集，能够有效地提高算法的计算效率[3]。

诸多近期的研究成果表明，基于元启发式的算法在求解聚类问题方面存在巨大的潜力。受到研究人员关注的元启发式算法主要有遗传算法(genetic algorithm, GA)[4]、差分进化算法(differential evolution algorithm, DE)[5]、蚁群算法(ant colony optimization, ACO)[6]、粒子群优化算法(particle swarm optimization, PSO)[7]等，这些元启发式算法均支持求解无监督聚类的问题。元启发式算法大多采用迭代寻优的策略，因此求解全局最优解的时间成本较高。许多研究者将 MapReduce 编程模型与元启发式算法结合，利用 MapReduce 的并行计算能力降低元启发式算法的总体处理时间。文献[8]成功地将 *k*-近邻算法在 MapReduce 模型实现，该方案明显地提高了 K-近邻算法对大数据集的处理效率。

目前，已有少量研究人员将元启发式算法在 MapReduce 编程模型上实现[9]，其中也存在一些针对聚类问题进行有效优化的方案。文献[10]提出了一种基于 MapReduce 和 K-means 的大数据聚类算法，该算法有效地在高维空间对大数据进行聚类处理，并且通过 MapReduce 框架实现算法的分布式处理，降低了内存成本与计算成本。文献[11]成功地将人工智能算法与分布式计算框架结合，在计算效率与聚类效果两方面同时取得了明显的效果；但是受限于遗传算法固有的不足，该方案对大数据的聚类准确率并未达到理想的效果。

人工蜂群算法(artificial bee colony, ABC)是一种全局搜索能力极强的人工智能算法，许多研究人员采用 ABC 已经成功解决了聚类问题[12]。虽然 ABC 的全局搜索能力强，但是 ABC 对于大规模问题的局部开发能力较弱。针对 ABC 局部开发能力弱的缺点，利用灰狼优化算法 (grey wolf optimizer, GWO) [13]增强 ABC 的局部开发能力，简称为 GWOABC(grey wolf optimizer artificial bee colony)算法。ABC 利用信息分享策略保持全局的搜索能力，通过 GWO 的捕食策略来增强局部的开发能力，该策略能够防止早熟收敛。此外，设计了基于混沌映射与反向学习算法的种群初始化机制，

收稿日期：2018-11-12；**修回日期：**2019-01-04 **基金项目：**湖北省教育厅科学技术研究重点项目（D20141005）

作者简介：孙倩（1980-），女，山东文登人，高级实验师，硕士，主要研究方向为信息安全、系统分析与集成(sunqianaro@126.com)；陈昊（1977-），男，教授，博士，主要研究方向为软件工程、智能计算；李超（1965-），男，湖北新洲人，高级实验师，主要研究方向为信息安全、计算机网络.

能够有效地解决随机初始化导致搜索性能不稳定的问题。并且基于 Hadoop 框架的 MapReduce 模型实现了 GWOABC 算法，通过分布式计算提高算法对于大数据的处理速度。

1　相关背景知识

1.1　灰狼优化算法(grey wolf optimizer, GWO)

GWO 算法[14]是一种模拟灰狼捕食行为的群智能优化算法，考虑了灰狼捕食行为的包围、追捕与攻击三个阶段。算法 1 所示是灰狼算法的伪代码。

算法 1 GWO 算法的伪代码

输入：搜索 agent 的种群 N，解的维度 n，解的上下界 $[Ub_1,\dots,Ub_n,Lb_1,\dots,Lb_n]$，最大迭代次数 $maxi$。

输出：最优 agent $\vec{X}_\alpha$。

1.初始化灰狼群 $\vec{X}_i=(x_1,x_2,\dots,x_n)$，其中($i\in[N]$)，$x_j\in[Ub_j, Lb_j]\,|\,\forall j\in[n]$。

2.初始化 α, $\vec{A}$, $\vec{C}$, t=1

3.计算每个搜索 agent 的适应度 $f(X_i)$，式中 $i\in[N]$

4. $\vec{X}_\alpha$=全局最优 agent；$\vec{X}_\beta$=第二优 agent；$\vec{X}_\delta$=第三优 agent；

5.while (t<$maxi$) {

6.　foreach agent {

7.　　更新当前 agent 的位置 /*根据(5)式计算*/

8.　}

9.　更新 α, $\vec{A}$, $\vec{C}$；

10.　计算所有 agent 的适应度；

11.　更新 $\vec{X}_\alpha$, $\vec{X}_\beta$, $\vec{X}_\delta$

12.　t=t+1;

13.}

GWO 模型主要包括了以下几个要素：

a) 社会等级。将三个最优解设为 α、β、δ 狼，剩下的狼群设为 ω 狼。

b) 包围猎物。包围猎物策略的数学模型表示为

$$\vec{D}=|\vec{C}\cdot\vec{X}_p(t)-\vec{X}(t)| \tag{1}$$

$$\vec{X}(t+1)=\vec{X}_p(t)-\vec{A}\cdot\vec{D} \tag{2}$$

其中：$\vec{A}$ 与 $\vec{C}$ 为两个系数向量，$\vec{A}=2\vec{a}\cdot\vec{r}_1-\vec{a}$，$\vec{C}=2\vec{a}\cdot\vec{r}_2-\vec{a}$。随机向量 $r_1,r_2\in[0,1]$，$\vec{a}=a_1(1-t/t_{max})$，一般将 a_1 值设为 2。t_{max} 为最大迭代次数。

c) 追捕阶段。三个最优解 α、β、δ 狼引导 GWO 的捕猎程序。ω 狼基于 α、β、δ 狼更新位置，位置的更新方法为

$$\vec{D}_\alpha=|\vec{C}_1\cdot\vec{X}_\alpha(t)-\vec{X}|,\quad \vec{D}_\beta=|\vec{C}_2\cdot\vec{X}_\beta(t)-\vec{X}|,\quad \vec{D}_\delta=|\vec{C}_3\cdot\vec{X}_\delta(t)-\vec{X}| \tag{3}$$

$$\vec{X}_1=\vec{X}_\alpha(t)-\vec{A}_1\cdot(\vec{D}_\alpha),\quad \vec{X}_2=\vec{X}_\beta(t)-\vec{A}_1(\vec{D}_\beta),\quad \vec{X}_3=\vec{X}_\delta(t)-\vec{A}_1(\vec{D}_\delta) \tag{4}$$

$$\vec{X}(t+1)=\frac{\vec{X}_1+\vec{X}_2+\vec{X}_3}{3} \tag{5}$$

d) 攻击猎物。参数 a 控制 GWO 的攻击猎物阶段程序，a 值随着迭代逐渐地减小。GWO 中两个参数 $\vec{A}$ 与 $\vec{C}$ 控制猎物的搜索过程，$\vec{A}$ 的取值为-2a~2a。如果 $\vec{A}$<1，那么灰狼攻击猎物。

e) 搜索猎物。参数 A 控制 GWO 的探索速度，如果$|\vec{A}|$>1，那么搜索的多样性较高。

1.2　ABC 算法

ABC 算法[15]主要由雇佣蜂阶段、观察蜂阶段、搜索蜂阶段三个阶段组成。雇佣蜂与观察蜂的搜索策略直接随机更新解向量的一个元素，如式（6）所示。

$$v_{ij}=x_{ij}+\phi_{ij}(x_{ij}-x_{kj}) \tag{6}$$

其中：v_{ij} 为两个不同解的第 j 维变量变异所获得的新解；ϕ_{ij} 为[-1,1]区间的随机数。

ABC 具有较强的全局搜索能力，但是 ABC 算法并未利用最优解来引导搜索程序，可能导致算法的收敛速度减慢。因为最优解信息能够提高算法的收敛性能，而 GWO 利用最优解信息的能力较强，所以将 GWO 引入 ABC 算法中，提高 ABC 算法的性能。

1.3　MapReduce 编程框架

为了提高 ABC 对大数据集的处理效率，本文设计了 ABC 在 MapReduce 模型的实现方案，简称为 MR-GWOABC。MR-GWOABC 处理大数据聚类任务的机制中包含两个主要的操作，即更新类的质心与适应度评估。类的质心更新基于 ABC 实现。适应度评估是计算每个目标与质心之间欧氏距离之和，并且寻找全局最优值。聚类程序将数据对象划分到各个簇中，最小化所有目标与质心之间的欧氏距离之和，将其作为 ABC 的适应度函数。基于 MR-GWOABC 的数据聚类流程如图 1 所示。

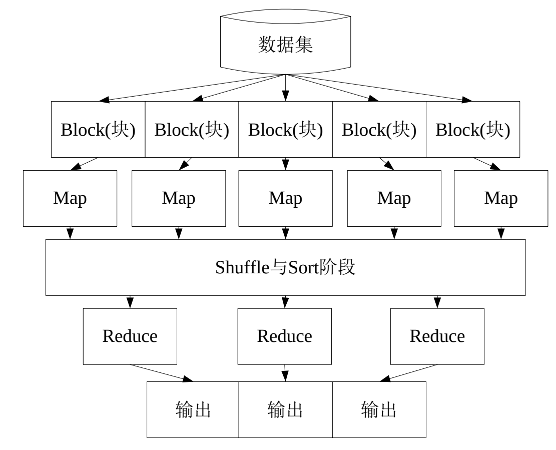


图 1　基于 MR-GWOABC 的数据聚类流程

Fig. 1　Data clustering flow based on MR-GWOABC

2　混合的 GWOABC 算法

文献[16]将传统 GWO 基于不同的标准 benchmark 函数进行了测试与验证，根据其分析结果，GWO 对于多模问题容易陷入局部最优解，原因在于种群的大部分灰狼仅仅作为一个跟随者，由三个最优解引导，导致整个种群的搜索能力较弱。

GWO 中参数 $\vec{a}$ 与 $\vec{A}$ 的值决定了种群全局搜索与局部开发之间的权重，如果 $|\vec{A}|$>1，那么候选解不同于依赖值；如果 $|\vec{A}|$<1，那么候选解收敛于依赖值。文献[16]发现 GWO 在早期迭代中搜索的能力较强，后期迭代中开发的能力较强。GWO 通过三个最优 agent 引导搜索的过程，导致算法缺少多样性。本文将 GWO 引入 ABC 算法，通过 ABC 的信息分享机制来提高候选解之间的信息分享，从而提高 GWO 的全局

搜索能力。

本文 GWOABC 算法的流程框图如图 2 所示。从图中可看出，GWOABC 的所有步骤与传统 GWO 相同，在种群初始化与信息分享模块中增加了一些附加策略。首先定义了初始化参数，包括种群规模 N、解空间维度 dim、最大的迭代次数；然后计算其他的参数，包括 a，A，C。算法主要分为三个阶段，分别为种群初始化阶段、GWO 阶段与 ABC 阶段。

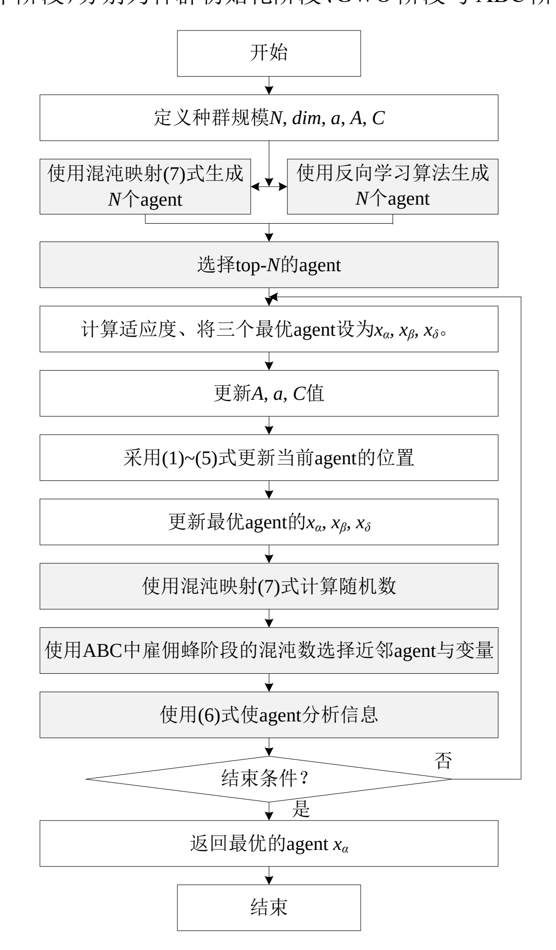


图 2　GWOABC 算法的流程

Fig. 2　Flow diagram of GWOABC algorithm

1)种群初始化阶段　通过混沌映射与反向学习算法生成优质的初始化候选解。算法 2 描述了种群初始化的策略。观察算法 1 的第 1 行与第 2 行，使用 Logistic 混沌映射处理获得随机变量 $ch(k)$，将结果作为初始化种群 X，将 X 定义为变量空间的边界。Logistic 混沌映射函数定义为

$$ch(k+1)=4*ch(k)*(1-ch(k)) \tag{7}$$

其中：k 为迭代次数，设为 300；初始值 $ch(0)$为随机值。

算法 2　基于反向学习与混沌映射的种群初始化程序

输入：搜索 agent 的种群规模 $|N|$，解的维度为 $j\in[n]$，解的上下界分别为 $x_j\in[Ub_1,\dots,Ub_n,Lb_1,\dots,Lb_n]$，$k$=3。

输出：初始化种群 $|N|$。

1.使用 n 维空间的混沌映射初始化种群 $|N|$，使用(7)式获得 $ch(k)$。

2.$X_i=(x_1,x_2,\dots,x_n)$ 的元素定义为 $x_j=Lb_j+ch(k)^*Ub_j-Lb_j$，其中 $i\in[N]$，$x_j\in[Ub_j, Lb_j]\ \forall j\in[n]$。

3.使用 OBL 获得种群大小为 N 的集合

4. $X_i^*=(x_1^*,x_2^*,\dots,x_n^*)$，其中 $i\in[N]$，每个新反向解定义为 $x_j^*=Lb_j+Ub_j-x_j$，$x_j^*\in[Ub_j,Lb_j\,|\,\forall j\in[n]]$。

5.将两组解结合 $X=(X_i\cup X_i^*)$；

6.计算适应度 $f(X)=(f(X_i)\cup f(X_i^*))$，并且按照适应度排序

7.选择 top-N 解。

第 4 行通过反向学习算法推导反向种群 X^*的集合，然后将两个集合组合为一个解 $X=(X_i\cup X_i^*)\in|2N|$，并且计算其适应度 $f(X)$。

2)GWO 阶段　生成初始化种群后，传统 GWO 使用式(1)~(5)更新其参数与当前 agent 的位置。

3)ABC 阶段　ABC 蜂群在候选解之间分享信息，根据式(6)更新旧解。为了提高序列的随机性、降低重复性，采用式(7)的 Logistic 混沌映射生成式(6)的 ϕ_{ij} 项。

迭代地运行 GWO 程序与 ABC 程序直至达到最大迭代次数，将最优解作为最终的结果。通过 ABC 的全局搜索改善了 GWO 的全局搜索能力，使得狼群的每个成员具备分享信息的机会；同时保留了算法的全局搜索与局部开发能力，有效地解决了多样性问题，并且防止早熟收敛问题。

3　基于 MR-GWOABC 的聚类算法

MR-GWOABC 在处理大规模数据的聚类任务中主要有两个模块，分别为更新质心处理与适应度评估。基于 MR-GWOABC 完成质心的更新处理，其目标是计算每个目标与质心之间的平方欧氏距离之和，获得全局最优值。聚类处理的目标是寻找数据实例的最优分配，实现平方欧氏距离之和的最小化，将该策略作为 ABC 的适应度函数。图 3 所示是基于 MR-GWOABC 的数据聚类流程。

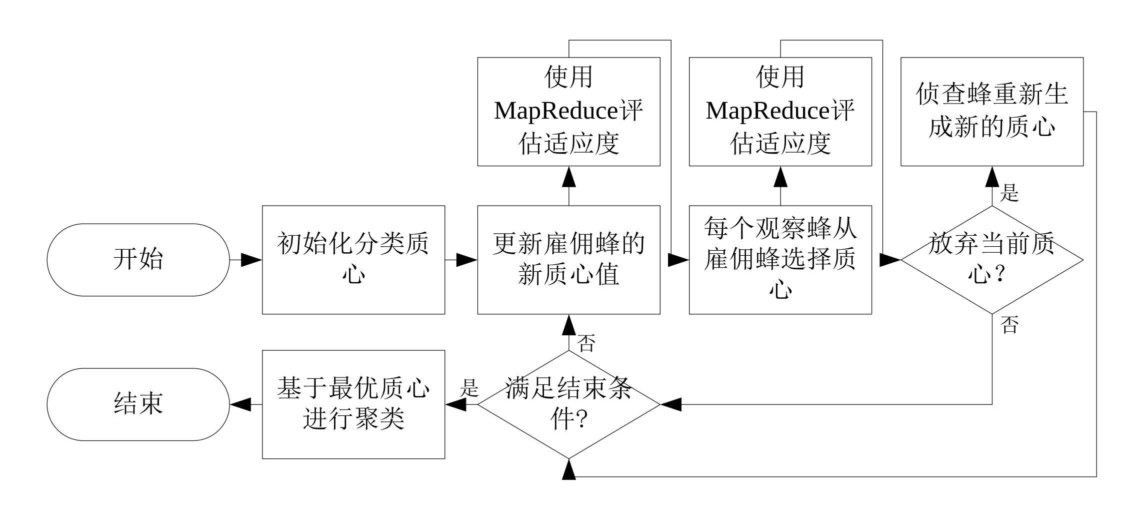


图 3　基于 MR-GWOABC 的聚类算法流程

Fig. 3　Flow diagram of MR-GWOABC based clustering algorithm

根据图 3 所示的流程框图，首先生成初始化解，将初始化解作为雇佣蜂的解。计算平方欧氏距离之和计算聚类的质量，评估解的适应度。雇佣蜂更新并与观察蜂分享它们发现的新解，然后观察蜂选择适应度较好的解。重复该程序，直至达到最大迭代次数。如果在一定时间内某个解的适应度无法提高，那么侦查蜂重新生成新解。然而大数据集的适应度评估需要大量的计算时间，所以采用 MapReduce 编程模型分布式地计算适应度。

算法 3、4 所示为 MapReduce 模型的 Map 函数与 Reduce 函数。Map 函数首先通过 GWOABC 算法检索聚类的质心，数据记录于 Hadoop 分布式文件系统(hadoop distributed file system, HDFS)中；然后 Map 函数提取每个蜂群的质心，计算每个数据与质心之间的距离，返回每个质心 ID 的最小距离。使用质心 ID 的最小距离建模 Map 函数的 key，从最小距离中获得一个新的 value。Map 函数将新的 key 与 value 传递至 Reduce 函数，Reduce 函数聚集所有相同的 key，计算总体的平均距离，最终，Reduce 函数 Map 函数调用 emit(key, value) 触发平均距离的 key，计算蜂群中每只蜂的适应度值。

算法 3 map 函数

```
输入：(key: record_id, value: record) 。
/*初始化*/
record_id = key;
record = value;
read(蜂群);
foreach bee in 蜂群 {
    bee_id = extract_bee_id(bee);
    CV = extract_centroids(bee);
    min_dist = return_min_distance(record, CV);
    centroid_id = i;
    new_key = (bee_id, centroid_id);
    new_value = (min_dist);
}
```

算法 4 reduce 函数

```
输入：key:(bee_id, centroid_id), value_list:(min_dist, 1)。
初始化：count = 0;
sum_dist = 0;
avg_dist = 0;
Foreach value in value_list {
    min_dist=extract_min_dist(value);
    count = count+1;
    sum_dist = sum_dst + min_dist;
}
avg_dist = sum_dist/count;
emit(key, avg_dist);
```

4　实验结果与分析

本文对 MR-GWOABC 算法进行仿真实验，评估聚类的质量与并行算法的有效性。实验中基于 Perl 脚本语言编程实现本算法，实验中 Hadoop 平台为 10 个节点，每个节点的配置为 CPU：2.26 GHz，4 GB 内存，180 GB 磁盘，每个节点的操作系统为 Ubuntu 14.04，Hadoop 平台为 Apache Hadoop 2.6.2。

为了基于大数据集评估 MR-GWOABC 算法的性能，采用表 1 所示的合成数据集。将四个公开数据集合成一个大数据集，四个数据集均来自于 UCI Machine Learning Repository，而且具有不同的属性。将四个数据集随机复制若干个备份，组成 10^7 个记录的大数据集。

表 1　实验数据集的属性

Table 1　Attributes of experimental datasets

编号	数据集	样本数量	数据维度	分簇数量
1	Iris	10000050	3	4
2	CMC	10000197	3	9
3	Wine	10000040	3	13
4	Vowel	10000822	6	3

4.1　聚类算法的性能

采用 F-measure 作为聚类质量的评估指标，F-measure 由精度与召回率两个信息指标计算而来，定义为

$$F(i,j)=\frac{2\cdot r(i,j)\cdot p(i,j)}{r(i,j)+p(i,j)} \tag{7}$$

其中：j 表示聚类算法生成的类；i 表示原数据集的类标签；r 与 p 分别表示召回率与精度。

召回率定义为 $r(i,j)=\frac{n_{ij}}{n_i}$，精度定义为 $p(i,j)=\frac{n_{ij}}{n_j}$。其中：$n_{ij}$ 表示类 i 的数据被分类为类 j 的数量；n_i 与 n_j 分别为类 i 与类 j 的数据规模。对于规模为 n 的数据集，总的 F-measure 计算公式为

$$F=\sum_i \frac{n_i}{n}\max_j(F(i,j)) \tag{8}$$

其中：F 的上界为 1，F-measure 值越大，聚类质量越高。

对于聚类之类比较，将本方案的解与 PKMeans 算法、并行 K-PSO 算法比较。三种聚类算法的性能结果如表 2 所示。MR-GWOABC 算法的性能明显地优于 PKMeans 与并行 K-PSO 算法。虽然三种聚类算法均采用基于启发式的搜索方法，但是 K-means 算法对于问题空间中初始化质心的位置较为敏感，K-means 算法趋向于收敛于起始点附近的局部最优解。将 ABC 算法与 PSO 算法比较，ABC 搜索的解质量优于 PSO 算法。ABC 方法的搜索过程中包含了全局搜索与局部开发两种策略，而 PSO 方法的搜索过程中则侧重于局部开发。在 ABC 算法中，雇佣蜂与观察蜂负责局部开发，侦查蜂负责全局搜索，如果搜索程序陷入局部最优，侦查蜂将随机搜索一个新的解。

表 2　三种聚类算法的 F-measure 结果

Table 2　F-measure results of three clustering algorithms

数据集编号	MR-GWOABC	PKMeans	K-PSO
1	0.842	0.667	0.785
2	0.387	0.298	0.324
3	0.718	0.482	0.517
4	0.643	0.586	0.627

4.2　聚类算法的时间效率

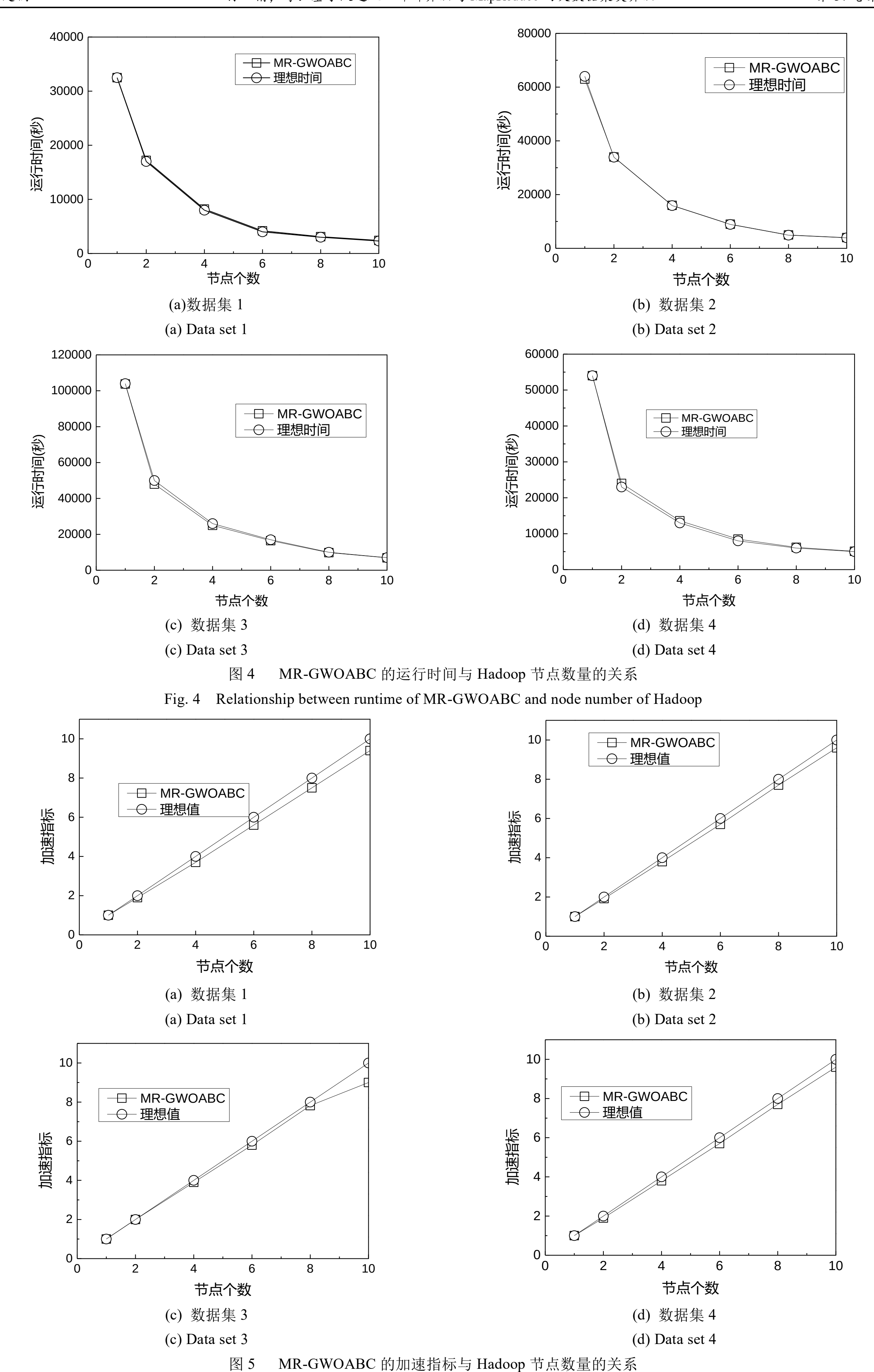
本算法的主要贡献是采用 MapReduce 模型提高了对大数据集聚类处理的速度，因此通过实验评估本算法的加速效果。将加速指标定义为 S_p，S_p 计算公式为

$$S_p=\frac{T}{T_N} \tag{9}$$

其中：T 表示一个 Hadoop 节点的处理时间；T_N 表示使用 N 个 Hadoop 节点的处理时间。

4.2.1 可扩展性实验

首先测试了 MR-GWOABC 算法的可扩展性，每次运行改变 Hadoop 节点的数量，统计本算法对于四个数据集的运行时间与加速指标，如图 4 与 5 所示。从图 4(a)~(d)可看出，MR-GWOABC 的运行时间随着 Hadoop 节点数量的增加而降低，并且运行时间接近于理想值。从图 5(a)~(d)可看出，MR-GWOABC 的加速指标呈现线性提高的趋势，并且其结果接近理想值。



(a)数据集 1

(a) Data set 1

(b) 数据集 2

(b) Data set 2

(c) 数据集 3

(c) Data set 3

(d) 数据集 4

(d) Data set 4

图 4　MR-GWOABC 的运行时间与 Hadoop 节点数量的关系

Fig. 4　Relationship between runtime of MR-GWOABC and node number of Hadoop

(a) 数据集 1

(a) Data set 1

(b) 数据集 2

(b) Data set 2

(c) 数据集 3

(c) Data set 3

(d) 数据集 4

(d) Data set 4

图 5　MR-GWOABC 的加速指标与 Hadoop 节点数量的关系

Fig. 5　Relationship between speedup of MR-GWOABC and node number of Hadoop

4.2.2 数据规模对可扩展性的影响

其次测试了数据集规模对于 MR-GWOABC 算法可扩展性的影响，效率的计算方法如式(10)所示。

$$效率=\frac{S_p}{N} \tag{10}$$

表 3 所示是 MR-GWOABC 在 10 个节点获得的效率结果，也是 MR-GWOABC 算法的性能随着数据集规模增大的变化情况。从结果可看出，不同规模数据集的效率均高于 0.90，并且可得出结论 MR-GWOABC 算法的可扩展性随着问题规模的增大而提高。

表 3 不同规模数据集的效率值与 F-measure 值

Table 3 F-measure results of efficiency values of different scale of datasets

数据集编号	性能指标	2 GB	4 GB	6 GB	8 GB	10 GB
1	F-Measure 值	0.842	0.842	0.842	0.842	0.842
	效率值	0.905	0.914	0.918	0.922	0.925
2	F-Measure 值	0.387	0.387	0.387	0.387	0.387
	效率值	0.924	0.927	0.93	0.935	0.941
3	F-Measure 值	0.718	0.718	0.718	0.718	0.718
	效率值	0.932	0.934	0.934	0.942	0.948
4	F-Measure 值	0.643	0.643	0.643	0.643	0.643
	效率值	0.911	0.918	0.924	0.927	0.928

本文的 MR-GWOABC 算法基于 MapReduce 实现，MapReduce 的 Map 任务将数据集转换为“键—值”对的数据集，Reduce 任务将这些数据组合成集合。MapReduce 分布式计算将任务分成若干个单独的进程，在大型硬件集群上并行地执行。MapReduce 将大数据集的各个元素分解为元组，然后进一步将元组缩小为较小的集合，通过将大数据数据并行地处理，由此显著地加快数据处理速度。通过实验结果可证明，MR-GWOABC 算法能够在合理的时间内完成大数据的聚类处理，并且保持较高的解质量。

5 结束语

为了提高聚类算法的处理效率与聚类性能，本文设计了基于改进人工蜂群算法与 MapReduce 的大数据聚类算法。将 GWO 算法与 ABC 算法结合，同时提高 ABC 算法的搜索能力与开发能力。该策略能够有效地提高聚类处理的性能。采用混沌映射与反向学习作为 ABC 种群的初始化策略，提高搜索的解质量。聚类算法基于 Hadoop 的 MapReduce 编程模型实现，通过最小化类内距离的平方之和，实现对大数据的聚类处理。实验结果表明，MR-GWOABC 算法能够高效地处理大规模数据集，并且实现了较好的聚类质量。

参考文献：

[1] 周润物，李智勇，陈少淼，等. 面向大数据处理的并行优化抽样聚类 K-means 算法 [J]. 计算机应用, 2016, 36 (2): 311-315. (Zhou Runwu, Li Zhiyong, Chen Shaomiao, *et al.* Parallel optimization sampling clustering K-means algorithm for big data processing [J]. Journal of Computer Applications, 2016, 36 (2): 311-315.)

[2] 海沫. 大数据聚类算法综述 [J]. 计算机科学, 2016, 43 (s1): 380-383. (Hai Mo. Survey of clustering algorithms for big data [J]. Computer Science, 2016, 43 (s1): 380-383.)

[3] 宋杰，孙宗哲，毛克明，等. MapReduce 大数据处理平台与算法研究进展 [J]. 软件学报, 2017, 28 (3): 514-543. (Song Jie, Sun Zongzhe, Mao Keming, *et al.* Research advance on mapreduce based big data processing platforms and algorithms [J]. Journal of Software, 2017, 28 (3): 514-543.)

[4] Ding Y, Fu X. Kernel-based fuzzy C-means clustering algorithm based on genetic algorithm [J]. Neurocomputing, 2016, 188: 233-238.

[5] 王勇臻，陈燕，张金松. 一种改进的求解聚类问题的差分进化算法 [J]. 计算机应用研究, 2016, 33 (9): 2630-2633. (Wang Yongzhen, Chen Yan, Zhang Jinsong. Improved differential evolution algorithm for clustering [J]. Application Research of Computers, 2016, 33 (9): 2630-2633.)

[6] Villar-Rodriguez E, Gonzalez-Pardo A, Ser J D, *et al.* A novel adaptive density-based ACO algorithm with minimal encoding redundancy for clustering problems [C]// Proc of IEEE Evolutionary Computation. 2016: 3139-3145.

[7] Santos P, Macedo M, Figueiredo E, *et al.* Application of PSO-based clustering algorithms on educational databases [C]// Proc of IEEE Computational Intelligence. 2018: 1-6.

[8] Anchalia P P, Roy K. The K-nearest neighbor algorithm using mapreduce paradigm [C]// Proc of IEEE International Conference on Intelligent Systems, Modelling and Simulation. 2015: 513-518.

[9] 于彦伟，齐建鹏，陆云辉，等. 时空轨迹大数据分布式蜂群模式挖掘算法 [J]. 计算机工程与科学, 2016, 38 (2): 255-261. (Yu Yanwei, Qi Jianpeng, Lu Yunhui, *et al.* Distributed swarm pattern mining algorithm in big spatio-temporal trajectory data [J]. Computer Engineering and Science, 2016, 38 (2): 255-261.)

[10] Tsapanos N, Tefas A, Nikolaidis N, *et al.* Efficient MapReduce kernel K-means for big data clustering [J]. Automatica, 2016, 43 (2): 1-5.

[11] Sinha A, Jana P K. A hybrid MapReduce-based K-means clustering using genetic algorithm for distributed datasets [J]. Journal of Supercomputing, 2017 (8): 1-18.

[12] 徐曼舒，汪继文，邱剑锋，等. 基于改进人工蜂群的模糊 C-均值聚类算法 [J]. 计算机工程与科学, 2016, 38 (6): 1238-1243. (Xu Manshu, Wang Liwen, Qiu Jianfeng, *et al.* A fuzzy C-means clustering algorithm based on improved artificial by colony [J]. Computer Engineering and Science, 2016, 38 (6): 1238-1243.)

[13] 徐辰华，李成县，喻昕，等. 基于 Cat 混沌与高斯变异的改进灰狼优化算法 [J]. 计算机工程与应用, 2017, 53 (4): 1-9. (Xu Chenhua, Li Chengxian, Yu Xin, *et al.* Improved grey wolf optimization algorithm based on chaotic Cat mapping and Gaussian mutation [J]. Computer Engineering and Applications, 2017, 53 (4): 1-9.)

[14] Mirjalili S, Mirjalili S M, Lewis A. Grey wolf optimizer [J]. Advances in Engineering Software, 2014, 69 (3): 46-61.

[15] Karaboga D, Basturk B. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm [J]. Journal of Global Optimization, 2007, 39 (3): 459-471.

[16] Kishor A, Singh P K. Empirical study of grey wolf optimizer [C]// Proc of the 5th International Conference on Soft Computing for Problem Solving. Singapore: Springer, 2016: 1037-1049.